Human behaviour recognition in data-scarce domains

Baxter, R. H., Robertson, N. M., & Lane, D. M. (2015). Human behaviour recognition in data-scarce domains. *Pattern Recognition*, *48*(8), 2377-2393. https://doi.org/10.1016/j.patcog.2015.02.019

Published in:
Pattern Recognition

Pattern Recognition 48 (2015) 2377–2393

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Human behaviour recognition in data-scarce domains

Rolf H. Baxter*, Neil M. Robertson, David M. Lane

Institute of Sensors, Signals and Systems, Heriot-Watt University, Edinburgh, EH14 4AS, UK

ARTICLE INFO

Article history:
Received 20 December 2013
Received in revised form
25 November 2014
Accepted 19 February 2015
Available online 6 March 2015

Keywords:
Behavior recognition
Bayesian inference
Visual surveillance
Behavior decomposition

ABSTRACT

This paper presents the novel theory for performing multi-agent activity recognition without requiring large training corpora. The reduced need for data means that robust probabilistic recognition can be performed within domains where annotated datasets are traditionally unavailable. Complex human activities are composed from sequences of underlying primitive activities. We do not assume that the exact temporal ordering of primitives is necessary, so can represent complex activity using an unordered bag. Our three-tier architecture comprises low-level video tracking, event analysis and high-level inference. High-level inference is performed using a new, cascading extension of the Rao–Blackwellised Particle Filter. Simulated annealing is used to identify pairs of agents involved in multi-agent activity. We validate our framework using the benchmarked PETS 2006 video surveillance dataset and our own sequences, and achieve a mean recognition *F*-Score of 0.82. Our approach achieves a mean improvement of 17% over a Hidden Markov Model baseline.

1. Introduction

Computer systems that can recognise human activity have captured the imagination of the research community for decades. This multi-faceted problem has drawn researchers from many different disciplines and has wide application potential including; systems that can monitor the wellbeing of people with a disability and the infirm (assisted living) [1], improving recognition in areas where human observers are suboptimal (e.g. security) [2], and improving situation awareness for autonomous vehicles [3].

Since video surveillance applications are the focus of our work, we begin with a motivating example. Fig. 1 shows two examples of the types of activity that occur in security footage (PETS2006). The 'Watched Item' activity represents two people travelling together where one traveller leaves their luggage in the custody of the other when he leaves the scene. The 'Abandon Item' activity is subtly different; it represents two travellers arriving *independently* but waiting in close proximity. In this circumstance, a person departing without their luggage is cause for concern while the former scenario is not.

The multi-disciplinary nature of activity recognition research has led to a set of terms that are inconsistent, and sometimes conflicting. In this paper behaviours and activities are synonymous, and will be discussed in two contexts; high-level (complex) and low-level (primitive). Primitive activities are isolated and do not involve long-term dependencies. Examples include 'enter-area' for somebody entering the field of view, and 'object-placed' for the placement of luggage on the floor by a person. Complex activities are composite and are comprised of sequences of primitive activities that achieve a higher-level goal. For example, watching a companion's luggage can be considered complex because it involves several components: some kind of association between people, placing luggage, the departure of one person and the monitoring of their luggage by another.

The relative infrequency of abandoned objects in security footage prevents statistically robust conclusions from being drawn via machine learning algorithms, which are becoming increasingly important for solving related computer vision problems. In automated surveillance it is not uncommon to manually specify semantic constraints [4,5], but these approaches are largely deterministic and lack convenient methods for handling observational uncertainty [1]. In the absence of training corpora one can rely on the mobilisation of human prior knowledge, but this too can be time consuming, expensive and unreliable [6]. In distributed, wide area surveillance, it is unclear if it would be possible to manually construct robust temporal models. Other state-of-the-art techniques for detecting object abandonment include monitoring the proximity of objects to their owners, although such techniques are unable to distinguish between the two motivating scenarios and do not generalise to other behaviours. Statistical and distance based anomaly detection algorithms are another two approaches for identifying irregular activity, but cannot be used to detect specific patterns of interest. Clustering techniques such as Hierarchical Dirichlet Processes [7] can be used to automatically

* Corresponding author. Tel.: +44 131 451 4168; fax: +44 131 451 4155.
E-mail addresses: R.H.Baxter@hw.ac.uk (R.H. Baxter), N.M.Robertson@hw.ac.uk (N.M. Robertson), D.M.Lane@hw.ac.uk (D.M. Lane).

http://dx.doi.org/10.1016/j.patcog.2015.02.019

discover activities from unlabelled training data, but cannot always distinguish between behaviours that are very similar [8].

The current state-of-the-art in video-surveillance research has failed to match the advances being made in plan recognition research, which has made significant advances in human activity recognition. The most robust plan recognition techniques adopt trained probabilistic models, although a limitation is that they are not well suited to data-scarce domains. By the phrase "data-scarce" we mean those scenarios where there is a natural lack of exemplars. In video surveillance applications in particular (since this is the application focus of this work) *accurately annotated* libraries of video do not exist for many of the interesting activities one would wish a machine to detect: anomalies and infrequently occurring activities.

This paper proposes that there is a better way to use probabilistic models in data-scarce domains and is fundamentally grounded upon the idea of an alternative activity representation that removes the need to learn temporal structure. We take our motivation from a phenomenon in psycholinguistics where randomising letters in the middle of words (or, 'radnsimnoig lteters in the mdidle of wrods) has little effect on the ability of skilled readers to understand the text [9]. We propose that like the letters in words, it is the primitive activities (e.g. items in ovals in Fig. 1) that are most important for allowing recognition, and de-emphasis the strict (temporal) ordering of those primitives. Specifically, we propose that the primitive activity subcomponents of a complex activity can be used as salient features, and that by imposing weak temporal constraints on the expected primitives we can recognise complex activities without learning their temporal structure. In doing so we are able to extend state-of-the-art techniques from plan recognition research to provide new algorithms for robust probabilistic recognition in data-scarce domains that are able to reason about uncertainty. The contributions of this work are:

1. A novel framework that builds upon existing research with Rao–Blackwellised Particle Filters by integrating a feature based representation and its Dynamic Bayesian Network implementation (thus retaining a unified and principled mathematical foundation). We demonstrate high recognition *F*-Score (0.82) in real-time within a noisy, sensor-based environment without model training.
2. We compare our approach against a set of Hidden Markov Model (HMM) classifiers, and show that our approach yields a mean improvement of 17% in *F*-Score.
3. Our method can recognise agents concatenating and switching between activities and remains robust to activities with significant similarities.
4. Inspired by cascaded Hidden Markov Models we develop a cascading particle filter to recognise activity at multiple levels of abstraction.

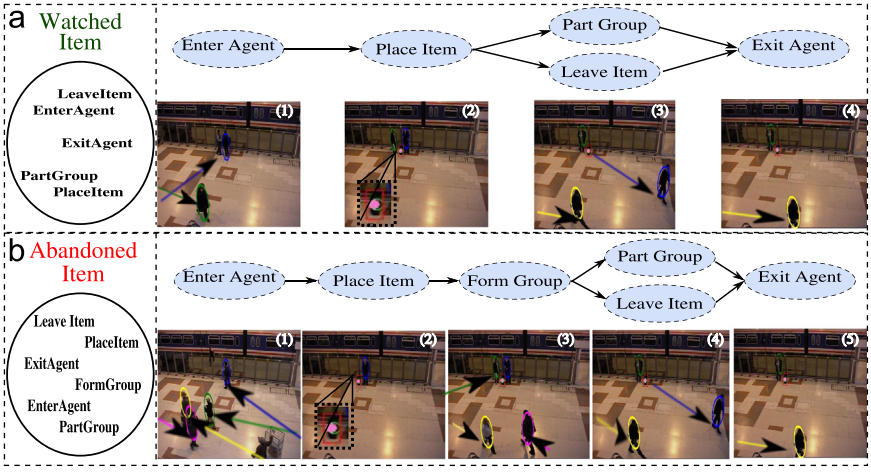

Fig. 1. Examples of the types of behaviours which occur in security footage (PETS2006) for which there are few training examples i.e. a "data-scarce" domain. We illustrate the real-time multi-person tracker employed by our system to recognise the primitive activities (e.g. 'Place Item') upon which higher-level (particle filter) inference is performed. Temporal ordering of activities is encoded only in the low-level video detectors.

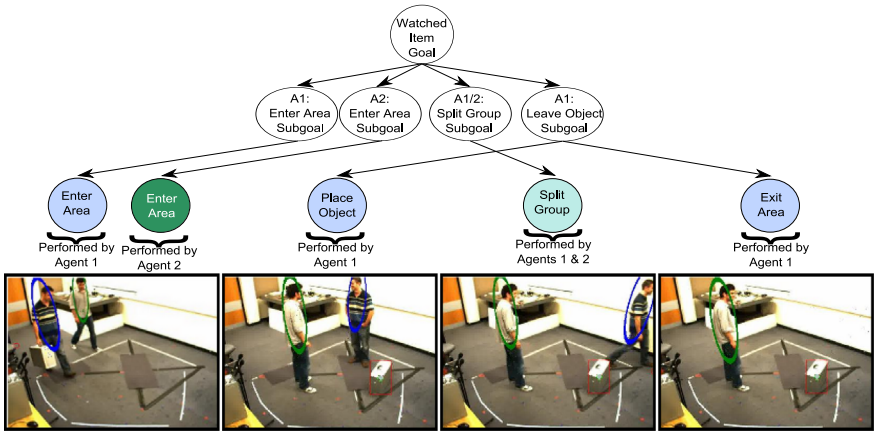

Fig. 2. An example of the composite nature of complex activity. At the bottom of the hierarchy primitive activities are detected directly from video. This complex activity has two distinct roles. (Video frames from our "HW" dataset.) (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

5. We achieve multi-agent (paired) activity detection by merging filtering densities from multiple particle filters and identifying the most probable joint activity explanation of all agents using combinatorial search, demonstrably improving the scientific state-of-the-art.

We validate our framework within the video surveillance domain using the PETS 2006 video surveillance dataset (e.g. Fig. 1), our own video sequences (e.g. Fig. 2) and a large corpus of simulated data.

The next section of this paper discusses related research. Sections 2 and 3 will provide an overview of our approach and encapsulates our ideas within a dynamic Bayesian network (DBN). Section 4 will show how efficient inference can be performed using a Rao–Blackwellised particle filter. Section 5 extends the representation into a hierarchical approach allowing multi-agent activities to be detected in Section 6. Section 7 will introduce the application domain and will describe the implementation details of our validation. We then discuss performance on both simulated and real video data and compare to the current state-of-the art before presenting conclusions and future work in Section 9.

1.1. Related work

Complex activity recognition: Recognising complex activities has primarily been the focus of plan recognition research. Early work often considered toy problems with manufactured observations [10], but an inability to reason about uncertainty prevented their application to real-world problems where observations are often noisy and uncertain (e.g. [11]). Links have also been drawn between activity recognition and natural language processing [12], although Nguyen et al. [13] highlighted that most of this work has focused on high-level inference and has not considered noisy observations from low-level sensors.

Many researchers have employed probabilistic techniques that use temporal information acquired through model training. This includes the Layered Hidden Markov Model by Oliver et al. [14] who demonstrated that multiple Hidden Markov Model (HMMs) could be employed in a tree-like structure. The most likely state from one level was cascaded into the next as a new observation to allow abstract recognition. Unlike our approach, Oliver performs inference on windows of observations rather than recursively.

Murphy [15] showed that HMMs are actually a special case of Dynamic Bayesian Network (DBN). DBNs also model temporal processes, but unlike HMMs, they can have any number of hidden states. Loy et al. [16] used a cascade of DBNs to model different temporal characteristics of activities. There are several significant differences from our work: (1) we employ a cascading structure for the purpose of activity abstraction and to model multi-agent activities, (2) Loy et al. detect anomalies instead of recognising specific activities.

Bui and Venkatesh [17], and Nguyen et al. [13,18] have shown that DBNs can model complex human activities. Recursive Bayesian estimation can perform efficient inference on DBNs and uses a factored distribution to update probability estimates as new observations arrive. Where exact inference becomes intractable Particle Filters can be employed to efficiently explore subregions of the state-space [19]. In this area the work by Bui and Venkatesh, and Nguyen et al., is the most relevant. Bui and Venkatesh used the Rao–Blackwellised Particle Filter [20] to recognise behaviours within an indoor environment of corridors and rooms. Unlike us, they segmented the environment into small cells to provide discrete agent locations and used trained trajectory models to predict agent movement. Nguyen et al. explored many of the same concepts using slightly different state models.

Recently, recognising complex activities in user generated video (e.g. YouTube) has gained increased attention. Merler et al. [21] use Support Vector Machines (SVMs) to learn primitive semantic classifiers (e.g. *beach*, *baseball*) using a large corpus of hand-labelled web images. By training a further set of SVMs on primitive activity feature vectors they are able to recognise three complex activities: *baking a cake*, *hitting a home run*, and *assembling a shelter*. In [22] Ma et al. use hand-labelled videos instead of web images to learn primitive video attributes. This allows them to learn spatio-temporal primitive activities such as *mixing cake batter*. As before, SVMs are used to learn the correlations between primitive and complex activities. A further example can be found in [23], this time combing SVMs with Deep Belief Networks [24]. In each of these cases, the approaches are well suited to multimedia video labelling, but require large labelled training corpora which makes them inappropriate for data scarce domains such as visual surveillance and security.

Multi-agent recognition: Recognising multi-agent activity can be broadly divided into two categories. The first considers physical groups of agents and often concerns their formation. Examples include recognising military formations and American football plays [25–27]. The second category recognises multiple agents performing different components of the same complex activity. Several approaches (e.g. [28,29]) have used strategies that look for the co-occurrence of activities, but assume that agents are members of a single team and do not consider environments with multiple independent agents/teams. Similar ideas are also encapsulated in [30], which implicitly recognises collaborative activities by ignoring action ownership.

Zhuo [31] adopts a constraint satisfaction approach using a MAX-SAT solver. They define activity pre-conditions and post conditions and are able to identify different agent teams collaborating towards joint goals. However, their approach is only applied to toy problems and is unable to deal with observation uncertainty.

Data scarcity: Progress in data-scarce domains has been limited. Security applications have failed to address a lack of training data [6], while video surveillance work has been limited to recognising primitive activities. For example, Tian et al. detect small objects as abandoned without considering whether they are attended [32]. Others have considered simple rules to detect that an agent has moved away from a bag they placed [33]. To our knowledge Ferryman et al. [5] is the only prior work to consider social relationships in object abandonment. Motivated by the 'Watched Item' scenario, they use the calibrated Social Force Model [34] to infer social groupings from trajectories, and combine this with heuristic rules to detect object abandonment. These rules are not probabilistic and only detect object abandonment. Rules have also been common in other vision research [30,35,4], but Lavee et al. [1] highlighted that they often lack the ability to reason about uncertainty.

Anomaly detection attempts to model *normal* activity for which training data is easier to obtain, so is well suited to data scarce domains. Anomalous activity is identified by virtue of its novelty, relative infrequency or distance from learnt models. A comprehensive introduction to anomaly detection can be found in Chandola et al. [36], and a more recent survey focusing on automated surveillance in Sodemann et al. [8]. Clustering algorithms are frequently applied to anomaly detection problems and context-driven approaches are particularly prevalent. Dee and Hogg [37] use Gaussian Mixture Models to identify pedestrian goal-locations from trajectories (e.g. entry/exit regions), and use manually specified obstacles stored in a polygon form to identify sub-goal locations. Tung et al. [38] remove the need for manually specifying obstacles by identifying turn points (e.g. obstacles), and model transitions between goal locations and turn points using a

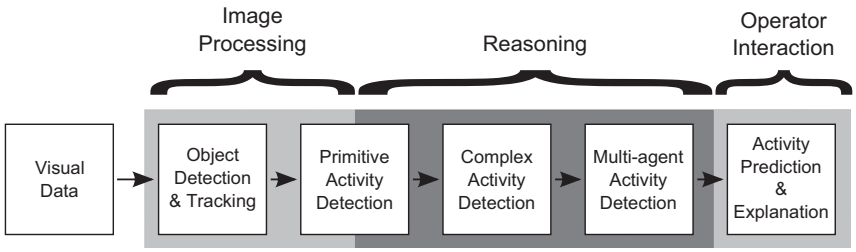

Fig. 3. A schematic of the main components of the system.

second-order Markov process. This allows sequential anomalies to be identified (e.g. irregular trajectory combinations) in addition to novel trajectories.

Arandjelović [39] divide trajectories into overlapping segments, identify the location and direction of each segment, and then perform clustering to identify *tracklets*: common motion primitives. A new track is expressed in tracklet models by divided it into overlapping segments, computing the direction and location as before, and associating each segment with the most similar tracklet primitive. They show two approaches for anomaly detection, the first using a first-order Markov model to model tracklet transitions, and the second using a distance-based approach. Several anomalous trajectories cannot be identified because of the inherent short-term memory of the first-order Markov model, while their distance-based model performs better.

Jiang et al. [40] remark that isolated trajectories are not always sufficient for anomaly detection, e.g. anomalies can arise due to inappropriate interactions between multiple targets. Their approach specifically addresses the problem of co-occurring object anomalies using an HMM to model normally co-occurring events.

Latent Dirichlet Allocation (LDA) [41], and the more recent (nonparametric) Hierarchical Dirichlet Process (HDP) [42] are two further algorithms that cluster data into different activities. An advantage of HDPs is that the number of activities does not need to be known a priori, while this is not the case for LDA. Niebles et al. [43] use LDA to recognise primitive activities such as hand-clapping and waving. Wang et al. [44] use HDPs for video based anomaly detection. A similarity with our approach is that these techniques adopt a bag-of-activities approach; that is, the temporal/spatial relationships between activities are neglected. However, a key difference from our work is that we only ignore temporal constraints in the representation, and impose weak constraints at the recognition stage. Moreover, a fundamental limitation of clustering techniques is that semantic labels cannot always be attached to the activity clusters identified, and rare or complex activities may not be clustered at all.

Many of the examples above are based on the assumption that anomalous activities are significantly different from normal activity. Sodemann et al. [8] highlight that a considerable limitation of these approaches is their inability to detect anomalies that are not significantly distinct from normal activity, and that this is particularly a problem for activities associated with planned crime and terrorism. It is this problem that we address here: our goal is to detect specific activities that may be very similar to normal activity and are therefore difficult to disambiguate (see example in Fig. 1). To that end, we use bags-of-activities because they allow an expert to define collections of specific activities of interest, whereas clustering techniques such as HDPs discover activities that are significantly different, and anomaly detection algorithms cannot be used to recognised specific activities.

HDPs have also been used for activity discovery and recognition outside of an anomaly detection context. For example, recent work by Phung et al. [7] used HDPs with unlabelled accelerometer data to discover activities such as sitting, walking and running. However, it has not been shown whether HDPs can automatically discover and distinguish between complex multi-person/multi-object activities such as those in this paper. As with other clustering techniques, this could be particularly challenging due to the similarities between some activities.

1.2. Summary of prior work

Prior work has shown that: (a) recursive Bayesian estimation is robust in noisy domains; (b) existing approaches often require large corpora either to learn the temporal structure of activities, or model normal activity to identify anomalies that are significantly different; (c) data-scarce domains have been deprived of these techniques; (d) approaches for recognising multi-agent complex activity have not been applied with open-world assumptions; (e) existing approaches only consider single-agent activity in a surveillance context and often rely on coarse heuristic rules (e.g. computing distances between person and object to infer "left luggage").

In the work of this paper, we present a new algorithm, which addresses these difficulties and validate it comprehensively on simulations and benchmarked datasets.

2. Overview of the approach

We show a schematic of the overall approach before explaining the components. Fig. 3 shows how each component of the work presented here depends on its predecessor. Although observations derived from the image processing inputs (these are primitive human activities) are the key to populating the data on which the reasoning layer operates, we first explain the inference engine. We then turn to the application domain and describe the tracker and primitive activity detectors.

We propose that complex activities can be modelled using unordered sets of primitive activities derived from video sequences. We do this by breaking the explicitly encoded temporal relationships between activities (which may themselves be primitive or complex). This is analogous to splitting pixel relationships for object detection and the model changes from "*Expect these ordered activities*" to "*Expect these activities*". This means that extensive corpora are no longer required to learn temporal structure. We return to Fig. 1a to illustrate this process. Fig. 1a showed two representations of the complex activity: 'Watched Item'. This activity involves two people travelling together, where one traveller leaves their luggage in the custody of the other while they leave the scene (extracting the primitive activities from video, which provide input to the model, is discussed in Section 7). For now we focus on the theoretical description of the Bayesian inference model. Illustrative frames of our tracked data are shown in Fig. 2. Coloured circles represent extracted activity primitives and can be easily observed using state-of-the-art video trackers and detectors. It will be seen when we discuss the video detectors in detail that some activities inherently retain a temporal

component, but the inference model does not rely on further temporal ordering, allowing activities to be observed, for example, from multiple views without a timestamp.

2.1. A motivating example

Because we ignore the temporal information, two complex activities may be indistinguishable in the special case that they consist of exactly the same primitives. However, this trade-off is balanced by the ability to recognise activities in domains where current techniques cannot be applied due to a lack of training data. Many complex activities do not create identical bags and we propose that as the size of complex activities increases the likelihood of generating identical bags reduces. Furthermore, in the special cases where identical bags are obtained, it is sometimes possible to alter the activities in the bags to allow disambiguation. To give a compelling example, consider the following two activity sequences:

1. Pickup Sandwich → Pickup Coffee → Pay → Leave (Normal)
2. Pickup Sandwich → Pay → Pickup Coffee → Leave (Potential theft of coffee)

In some cases encoding temporal information might be overly restrictive. For example, some establishments expect payment for an item before the item is picked up (e.g. coffee shop), while elsewhere this might not be allowed (e.g. cafeteria). We propose that an alternative representation would be to consider that customers pay, while thieves do not. It is more informative to model the need for a 'Pay' activity, rather than focusing on ordering. This allows modelling the same two activities like so:

1. {Pickup Item, Pay, Leave} (Normal)
2. {Pickup Item, Leave} (Theft)

With these new bags-of-activities we can clearly distinguish between theft and normal, and have also removed the constraint of requiring a rigid temporal structure that over-fits the model to a particular scene.

Rather than using a strict sequential model to predict observations for a complex activity B^i, our new model monitors the intersect between the bags of observed and expected activities. If, like [45], we make the simplifying assumption that each activity is only generated once per complex activity, then the set of expected activities are the elements of B^i not yet observed. Perhaps surprisingly, for the benchmarked data which is of most interest to the surveillance community this assumption holds well (as is demonstrated by the use of public datasets for primary validation). We address extensions to this work in Section 9 which may be useful in the cases where this assumption does not hold. To clarify, this assumption does not say that activities cannot be detected twice through mis-detection, and similarly, repetition of an entire complex activity does not contravene the assumption. If this set of expected activities is continually updated we can apply a weak temporal ordering to the elements of B^i. Although simplistic, this replacement temporal structure is adequate for recognising all of the activities within our validation set, while remaining weak enough to allow novel activity permutations to occur.

Let us define C as the set of currently observed activities and T as the set of target activities for B^i. We make the simplifying assumption that the activities of T can be defined by a domain expert. $T \setminus C$ is the set of expected (future) activities. Denote α^i as the ith activity from the set of detectable activities. At each time step activities in $T \setminus C$ have uniform probability, while all other activities have zero probability.

2.2. Worked example

Using the 'Watched Item' example from Fig. 1a, at time step $t=0$ each of the 5 activities has equal probability and $C = \varnothing$. In this example $P(EnterAgent) = 1/5$. Let us assume that *EnterAgent* is observed at $t=1$ and thus at $t=2$, $C = \{EnterAgent\}$. Because of the single-occurrence assumption $P(EnterAgent) = 0$ at $t=2$ while the elements of $T \setminus C$ have uniform probability: $\forall i \in T \setminus C : P(i) = 1/4$. This process repeats until all elements of T have been observed. Note that any element that is not in T has a zero probability at all time-steps. Furthermore, when all elements of T have been observed no further activities are expected.

3. Basic representation

Motivated by previous work modelling complex human activity (e.g. [17,13,18]), we encapsulate the bag-of-activities approach using a Dynamic Bayesian Network (DBN): an acyclic graphical model where each time-slice denotes the state of a system as a snapshot and each node denotes a state variable. Directed edges between nodes represent dependencies, while the absence of an edge implies conditional independence. DBNs allow us to utilise a large class of algorithms developed for Bayesian inference, including the ability to reason about uncertainty and perform recursive Bayesian estimation (discussed in more detail later).

Fig. 4 shows a two-slice DBN for our approach. To specify a DBN using the notation from [15], assume that Z_t represents all of the nodes in a given time-slice. A DBN is then defined to be a pair $(B_0, B_{\rightarrow})$, where B_0 is a Bayes Net defining the prior $P(Z_1)$ and $B_{\rightarrow}$ is a two-slice Bayes Net defining $P(Z_t | Z_{t-1})$ such that

$$P(Z_t | Z_{t-1}) = \prod_{i=1}^{N} P\left(Z_t^i | Pa(Z_t^i)\right) \quad (1)$$

where $Pa(Z_t^i)$ are the parents of node Z_t^i. Note that DBNs assume that the process being modelled is Markov and thus the state at time t is only dependent on the state at the previous time step: $P(Z_t | Z_{1:t-1}) = P(Z_t | Z_{t-1})$.

Returning to Fig. 4, nodes T and C represent the target and currently observed activity sets as before, respectively. Adopting terminology from the belief-desire-intention (BDI) architecture of human reasoning [46], node D represents the next desired activity and is conditionally dependent upon the target and currently observed activity sets. Node A represents the activity detected and is dependent upon the agent's desire. Moving to the top of the DBN, node I represents an activity interruption. We will discuss

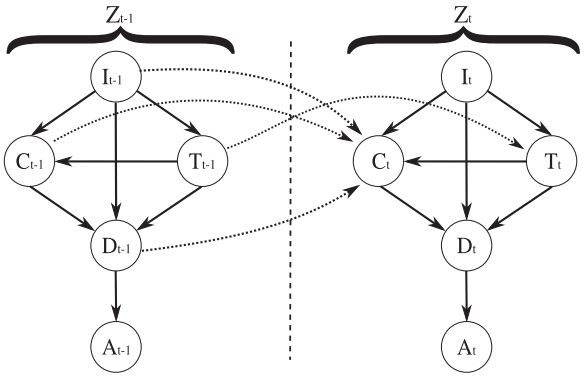

Fig. 4. The two-slice Dynamic Bayesian Network for the Bag-of-Activities.

Table 1
Prior, Conditional and Transition probabilities between time steps $t-1$ and t for the DBN in Fig. 4. Node I has three possible values: *FP* (False positive), *TPS* (True Positive Same), and *TPCh* (True-Positive Change).

	Priors: $P(C_1=\varnothing)=1,\ P(I_1=TPS)=1,\ P(T_1=B^i)=\lvert B\rvert^{-1}$		
1	$P(T_t=T_{t-1}\mid I_t=TPS)$	$=1$	Same complex activity
2	$P(T_t=B^i\mid I_t=TPCh)$	$=\lvert B\rvert^{-1}$	Complex activity changed
3	$P(T_t=T_{t-1}\mid I_t=FP)$	$=1$	Misdetection
4	$P(C_t=C_{t-1}\cup D_{t-1}\mid I_{t-1}=TPS, D_{t-1}=A_{t-1})$	$=1$	Previous desire was observed
5	$P(C_t=\varnothing\mid I_t=TPCh\cap\lvert C_{t-1}\rvert=\lvert T_{t-1}\rvert)$	$=1$	Target set changed, prev. beh. completed
6	$P(C_t=\{C_{t-1}\cup D_{t-1}\}\in T_t\mid I_t=TPCh)$	$=1$	Target set changed and $D_{t-1}=A_{t-1}$
7	$P(C_t=C_{t-1}\in T_t\mid I_t=TPCh)$	$=1$	Target set changed and $D_{t-1}\neq A_{t-1}$
8	$P(C_t=C_{t-1}\mid I_{t-1}=FP, I_{t-1}=TPS)$	$=1$	Previous Obs. was a mis-detection
9	$P(D_t=\alpha^i)$	$=\lvert C_t\backslash T_t\rvert^{-1}$	$\forall_i:\alpha^i\in C\backslash T$
10	$P(D_t=\alpha^i)$	$=0$	$\forall_i:\alpha^i\ni C\backslash T$
11	$P(D_t=\alpha^i\mid C_t,T_t,I_t=FP)=\lvert\alpha\rvert^{-1}$	$\forall_i:\alpha^i\in\alpha$	

how interruptions are detected in due course, but for the discussion at hand we can summarise that interruptions occur for two reasons:

- When an agent changes their complex activity the target activity set changes. (I=*TPCh* (*T*rue-*P*ositive *Ch*ange))
- When an activity is misdetected the agent's desire is not constrained to $T\setminus C$. (I=*FP* (*F*alse-*P*ositive)). Traditionally, false positive (FP) detections are handled via $P(A\mid D)$. However, in this work we also considered agents that switch and concatenate complex activities, both of which could result in observations that are misconstrued as FP detections. Consequently, we model all three conditions using node I.

When an interruption has not occurred I=*TPS* (*T*rue-*P*ositive *S*ame). If variables $\{C,T,I,D\}$ are referred to as latent variable Z, and the observation A as y then the network is reduced to the generic form: $Z\rightarrow y$ giving rise to what is commonly referred to as the observation model: $P(y\mid Z)$.

To fully specify the DBN the prior, conditional and transitional dependencies must be defined. This is best achieved with the descriptions below and the formal notation in Table 1.

1. When there is no interruption ($I_t=TPS$) the target activity set remains the same.
2. When a change of complex activity is detected ($I_t=TPCh$) the target activity set is re-initialised according to the uniform distribution.
3. When a misdetection occurs, the target activity set remains the same.
4. The set of currently observed activities (C_t) gains the agent's last desire (D_{t-1}) when it matches the last observation (A_{t-1}) and there was no interruption.
5. The set of currently observed activities (C_t) becomes the empty set when the target set has changed and all elements of T_{-1} had been observed.
6. When the target set has changed ($I_t=TPCh$) the set of currently observed activities becomes the elements of $C_{t-1}\cup D_{t-1}\in T_t$ when $D_{t-1}=A_{t-1}$. Note that C_t retains the elements from $C_{t-1}\in T_t$ to facilitate switching from one complex activity to another.
7. When the target set has changed ($I_t=TPCh$) the set of currently observed activities becomes the elements of $C_{t-1}\in T_t$ when $D_{t-1}\neq A_{t-1}$.
8. The set of currently observed activities (C_t) remains the same as (C_{t-1}) when the previous time-step was a misdetection ($I_{t-1}=FP$).
9. The desire(D_t) has a uniform probability for all elements in $C\setminus T$.
10. The desire (D_t) has a zero probability for any element not in $C\setminus T$.
11. The desire (D_t) has uniform probability for all elements in α when I=*FP*.

4. Basic inference

We will first explain how inference is performed on our flat activity structure before introducing a hierarchical algorithm that allows activity abstraction and multi-agent activities to be modelled. To achieve real-time recognition approximate inference is performed, while online recognition is achieved by using a recursive algorithm that updates probability estimates as observations arrive. Our inference algorithm is based on the Rao–Blackwellised particle filter (RBPF) [20,17], which we briefly describe below before applying to our model.

4.1. Particle filtering

Particle filtering can be used for approximate recursive Bayesian estimation. Using the general state-space model with hidden variables Z_t and observed variable y_t the goal of filtering is to calculate $P(Z_{1:T}\mid y_{1:T})$, or more frequently, the filtering distribution: $P(Z_T\mid y_{1:T})$. We represent this distribution using a set of N weighted samples (particles) denoted $\{Z^i_{1:T},\omega^i_T\}^N_{i=1}$. Let $Z^i_{1:T}$ be the i'th particle sampled from $P(Z_{1:T})$, and ω^i_t its associated weight such that $\sum_{i=1}^N\omega^i_t=1$ and $\omega^i_T=\prod_{t=1}^T P(y_t\mid Z_t)$. From this sample we approximate the filtering distribution via:

$$P(Z_t\mid y_{1:t})\approx\sum_{i=1}^N\omega^i_t\delta\left(Z_t,Z^i_t\right) \tag{2}$$

The general particle filtering algorithm is often referred to as Sequential Importance Sampling with Resampling. Full details can be found in [47], but to summarise, the particles are initialised by generating $\{Z^i_1,\omega^i_1\}^N_{i=1}$ from the prior and assigned uniform weights. The algorithm then proceeds as follows:

- *Sample*: N times with replacement from $\{Z^i_{t-1},\omega^i_{t-1}\}^N_{i=1}$.
- *Transition*: Draw $Z^i_t\sim Q(Z^i_t\mid Z^i_{t-1},y_t)$.

- *Weight*: Update the particle weight such that

$$\omega_t^i = \omega_{t-1}^i \frac{P(y_t|Z_t)P(Z_t|Z_{t-1})}{P(Z_t|Z_{t-1},y_t)} \tag{3}$$

- *Re-sample*: Generate a new set of particles $\{Z_t^{i*},\omega_t^{i*}\}_{i=1}^N$ by sampling N times with replacement from the approximate representation of $P(Z_T|y_{1:T})$ so that $P(Z_t^{i*}=Z_t^i)=\omega_t^i$. Set new weights to: $\{\omega_t^{i*}=1/N\}_{i=1}^N$.

The effect of re-sampling is that particles with very low weights in $\{Z_t^i\}_{i=1}^N$ are less frequent in $\{Z_t^{i*}\}_{i=1}^N$, while particles with higher weights are more frequent.

4.2. Rao–Blackwellisation

The purpose of Rao–Blackwellisation is to reduce the size of the sampled state-space to increase efficiency [20]. This is achieved using a model's structure to analytically marginalise some variables conditioned upon others. To proceed, partition the variables in Z_t into those that will be sampled, and those that will be marginalised. Denote the sampled component as r, and the marginalised component $z: Z_t=\{r_t,z_t\}$. The posterior filtering density can then be expressed by the following factorisation:

$$P(Z_t|y_{1:t}) = P(y_t|z_t,r_t)P(z_t|r_t)P(r_t|y_{1:t-1}) \tag{4}$$

In the next subsection we show how this factorisation is applied to our model.

4.3. Application to our model

Eq. (4) is a filtering distribution, and can be approximated using a Rao–Blackwellised Particle Filter (RBPF) by partitioning the DBN in Fig. 4. Returning to the model's structure, observe that variable D_t is conditionally dependent upon $\{C,T,I\}_t$. Recall that D_t represents the agent's desire: the next activity to be performed, and is trivial to calculate conditioned upon $\{C,T,I\}_t$. This dependency structure is particularly useful for Rao–Blackwellisation, which can be applied by segmenting the latent variables such that $r_t=\{C,T,I\}_t$ and $z_t=\{D\}_t$. The RBPF therefore consists of N random samples of the form $\{\{r,z\}_{1:t}^i,\omega_t^i\}_{i=1}^N$ that characterise the posterior density $P(Z_t|y_{1:t})$, where each sample point $\{r,z\}_t^i$ has an associated weight ω_t^i such that $\sum_{i=1}^N \omega_t^i=1$. To approximate the posterior density at time t:

$$P(Z_t|y_{1:t}) \approx \sum_{i=1}^N \omega^i \delta\left(\{r,z\}_t,\{r,z\}_t^i\right) \tag{5}$$

$$P(Z_t|y_{1:t}) \approx \sum_{i=1}^N \omega^i \delta\left(\{C,T,I,D\}_t,\{C,T,I,D\}_t^i\right) \tag{6}$$

The dependencies in (3) are too complex to calculate accurately without model learning. Our approach therefore applies the principles of particle weighting to generate a weight. It is composed of two factors: $P(Z_t^i|y_t)$: the true positive probability of the observation and the proportion of activities in the target set that have been observed. These components can be combined as follows:

$$\omega_t^i = \omega_{t-1}^i \times P(Z_t^i|y_t) \times \frac{|C_t^i|+1}{|T_t^i|+1} \tag{7}$$

A particle will attract a weight of zero whenever it cannot explain y_t. To prevent filter collapse (all particles having zero weight) a particle regeneration step is applied to detect and re-initialise such particles after the sampling step. This can be achieved by identifying where $P(A_t^i|\{C,T,I\}_t^i,A_{1:t-1}^i)=0$ and adding such particles to regeneration set $\Re$. The remaining particles form the eligible set ℓ such that $\{\Re_t\cup\ell_t\}=\{Z_t^i\}_{i=1}^N$ and $\{\Re_t\cap\ell_t\}=\emptyset$.

A subset of $\Re$ proportional to $1-P(D_t|A_t)$ are selected to represent a false-positive observation and are assigned a nominal weight of 0.01 (arbitrarily chosen and held static throughout the validation). The remaining particles in $\Re_t$ are used to assume that the agent has changed their complex activity, and thus variable I is set to *TPCh* and the variables are re-initialised per Table 1. Algorithm 1 summarises our inference procedure.

Algorithm 1. The basic RBPF inference algorithm.

```
Init: Generate [{{C,T,I}_0^i, ω_0^i}_{i=1}^N] ~ P(Z_1) and ω_1
for t=1 to T do
  for i=1 to N do
    Sample {C,T,I,D}_{t-1}^i
    Transition {C,T,I,D}_{t-1}^i to obtain {C,T,I}_t^i
    if P(A_t|{C,T,I}_t^i)=0 then
      if random() < 1−TP(A_t) then
        Set I=FP and weight 0.01 //False-positive
      else
        if C_t^k = T_t^k then
          Reset with prior // complex activity completed
        else
          Reset for activity change
        end if
      end if
    end if
    if I_t^i ≠ FP then
      Calculate RB-Posterior: P(D_t^i|{C,T,I}_t^i, A_{1:t})
      Predict D_t^i from RB-Posterior giving {C,T,D,I}_t^i
      Weight with (7)
    end if
  end for
  Normalise weights and re-sample Z_t
end for
```

5. Hierarchical representation

A more natural representation of activity is to consider a hierarchical decomposition. See for example the hierarchical representations used in the literature [17,13,18]. Updating the algorithm to incorporate hierarchical structure achieves the following:

1. Allows multi-agent activity to be modelled.
2. Aids activity specification via re-usable components.
3. Facilitates the explanation of recognised activities.

To adapt the algorithm into a hierarchical model the representation is first set into a hierarchical context. To do so, we continue to use the terminology of primitive and complex activities, but also introduce the term *root activity* to refer to the most abstract complex activity in a hierarchy. Recall that primitive activities are short term and achieve an immediate goal, while complex activities are composed of a number of primitive and/or complex activities. From these definitions one can immediately construct tree structures such as Fig. 2, where PlaceObject would be considered a primitive activity and LeaveObject would be considered complex. In this example 'Watched Item' is the root activity.

With this representation hierarchical recognition is achieved by considering each layer of complex activities as a set of alternative hypotheses. The target set for each complex activity is derived from its child components and thus the root activity 'Watched Item' has four components. In the flat model desires had uniform probability

over the set $C \setminus T$, however, this is now updated to be proportional to the number of children that comprise the activity. As an example of this updated distribution, $P(D_t = \text{LeaveObject} \mid \{C, T, I\}_t) = 0.4$ while $P(D_t = \text{SplitGroup} \mid \{C, T, I\}_t) = 0.2$ when $C = \varnothing$ and $I = TPS$.

Unlike some approaches (e.g. [13]), we do not need to model a termination distribution for each activity for three reasons: 1) we assume that all child activities terminate immediately, 2) the hierarchy is kept in lock-step, preventing transitions at one layer without the others, and 3) a complex activity will have zero probability of further execution once all child activities have been observed. In essence, this means that each layer of the hierarchy is decoupled and can be recognised independently using the flat inference algorithm already introduced. To recognise a set of hierarchical activities with depth $\tilde{D}$ thus requires a particle filter for each layer of complex activities forming a collection of $\tilde{D}-1$ filters.

For a set of complex activities at level l, a single particle filter approximates $P(Z_T^l = \alpha_l^i \mid y_{1:T})$, and thus we can recursively calculate $P(Z_T^l = \alpha_l^i \mid y_{1:T})$ for any level by starting at the bottom of the hierarchy $(\tilde{D}-1)$ and cascading up filtering densities to approximate $P(Z_T^l = \alpha_l^i \mid y_t)$ at the level above.

6. Multi-agent recognition

A benefit of the hierarchical filter is that it can model multi-agent activity. This is achieved by assigning agent-roles to complex activities with the implication that if activities γ and ι are assigned different roles then they must be performed by different agents. Continuing with the example from Fig. 2 the agent fulfilling role 1 *enters* the scene, *places* an object, *splits* from an agent group and *exits* the scene. Agent 2 *enters* with Agent 1, *splits* from the group and stays in the scene. Assigning roles to complex activities has minimum impact on the DBN, with the only notable change being that role information needs to be present within both T and C and the agent assigned to a role needs to be tracked to ensure consistency.

The addition of multi-agent activity converts the task into a multi-target tracking problem, where the number and identity of targets is unknown. Our solution to this problem is based upon combinatorial optimisation under the premise that the most likely activities for a set of agents can be used to identify those involved in multi-agent/solo activity. To maintain tractability we restrict multi-agent activity to pairs of agents and make the assumption that agents can only exhibit one root activity at a time.

Because filters must only receive observations for the agent (s) being tracked multi-agent and solo activities must be filtered independently. To determine whether an agent is involved in solo or multi-agent activity thus requires posterior densities from multiple filters to be combined to give a single, normalised filtering density. This is required for each potential pair of agents.

6.1. Merging filter posteriors

There are several stages to merging filter posteriors.

6.1.1. Calculate the un-normalised weight of each filter

Denote M and S as a multi-agent and solo filter respectively. To preserve notation similarity denote $P(S_t \mid y_{1:t})$ as the posterior probability of solo activities and $P(M_t \mid y_{1:t})$ as the posterior for multi-agent activities. Where as a normalised particle weight is denoted ω^i, let ω^{i*} denote the un-normalised particle weight. And if $f \in \{S, M\}$ is a filter representing the solo or multi-agent activities, and B^f is the set of associated root activities, then f_t^i is the i'th particle in filter f and $\delta(f_t^i, B_b^f) = 1$ when particle i represents the b'th activity in B^f. The un-normalised weight (FW^f) of filter f is then given by

$$FW^f = \sum_{b=1}^{|B^f|} FW_b^f = \sum_{b=1}^{|B^f|} \sum_{i=1}^{|f|} \omega^{i*} \delta(f_t^i, B_b^f) \tag{8}$$

where $|f|$ indicates the number of particles in filter f.

6.1.2. Calculate the importance of each filter

To re-normalise the posterior density estimates the importance of each filter is required. For example, if $FW^S = 100$ and $FW^M = 50$ then the importance of each filter can be described as $\text{Imp}(FW^S) = 100/150 = 0.67$ and $\text{Imp}(FW^M) = 50/150 = 0.33$. However, it should be noted that this calculation is only correct if the number of particles in each filter is equal. If the number of particles is imbalanced then the filter weights must first be equalised by normalising by $|f|$: The number of particles in each filter. Denote $Eq()$ as a function to perform this.

6.1.3. Re-normalise the posteriors

The final step is to recompute the likelihood of each activity B_b^f by weighting it with the filter's importance:

$$P(Z_t = B_b^f \mid y_{1:t}) \approx \text{Imp}(Eq(FW^f)) * P(f_t = B_b^f \mid y_{1:t}) \tag{9}$$

Given a set of combined solo/multi-agent filtering densities for a set of B root activities and N agents, combinatorial optimisation can be used to identify the most likely joint distribution. However, such a process requires $O(B^2N^2)$ operations. In order to achieve real-time recognition we reduce this runtime by opting for a heuristic approach based on simulated annealing [48]. We represent the joint root activity probability as a solution's utility and thus the objective of simulated annealing is to maximise the utility.

Because the optimal assignment of agents to root activities changes with each observation we make predictions when root activity termination is detected. This can be achieved by analysing the particles that represent the optimal root activity assignments. If the majority of those particles have observed all features (i.e. $C_t = T_t$) then the activity has terminated and a prediction can be made.

7. Video surveillance application

To validate our algorithm in a realistic domain we implemented a visual surveillance application using object detection, tracking, and event recognition techniques. Visual surveillance is one area in which annotated corpora are rarely available due to privacy concerns and the time consuming nature of annotating video. As stated in Section 1, the majority of existing research in this area has focused on recognising primitive activities and where complex activity has been considered, non-probabilistic event matching techniques are prevalent. Vision algorithms were used to extract object tracks from video sensor data, from which primitive activities could then be detected. These primitives were then provided as input to the reasoning (bag-of-activities) recognition algorithm. For the purposes of reproducibility, we briefly describe the implementation of our three-layer framework which consisted of: Image Processing, Reasoning, and Operator Interaction. Within each layer different processes were performed while individual layers were strictly de-coupled to facilitate the integration and replacement of different processing techniques. The overall framework is shown in Fig. 3.

The Image Processing layer detects the essential primitive activities for the RBPF algorithm, and is described in the following section. Simulations were used to validate the method prior to deploying a real-time tracker. A schematic outlining the contribution of these and the primitive activities extracted is shown in Fig. 5.

At the Operator Interaction layer activity predictions were made whenever a root activity completed all child activities. For

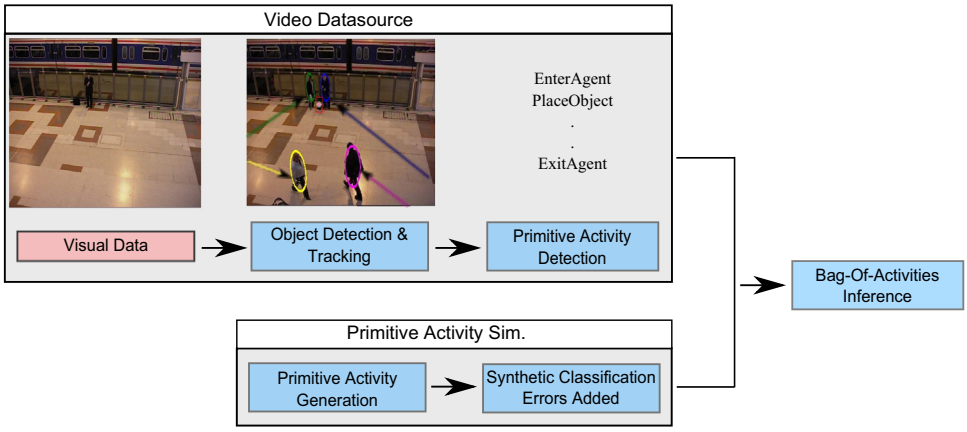

Fig. 5. The video processing components of the framework provide a stream of primitive activity observations to the bag-of-activities inference algorithm. The primitive simulator also provides such a stream and incorporates synthetic classification errors to mimic video processing failures.

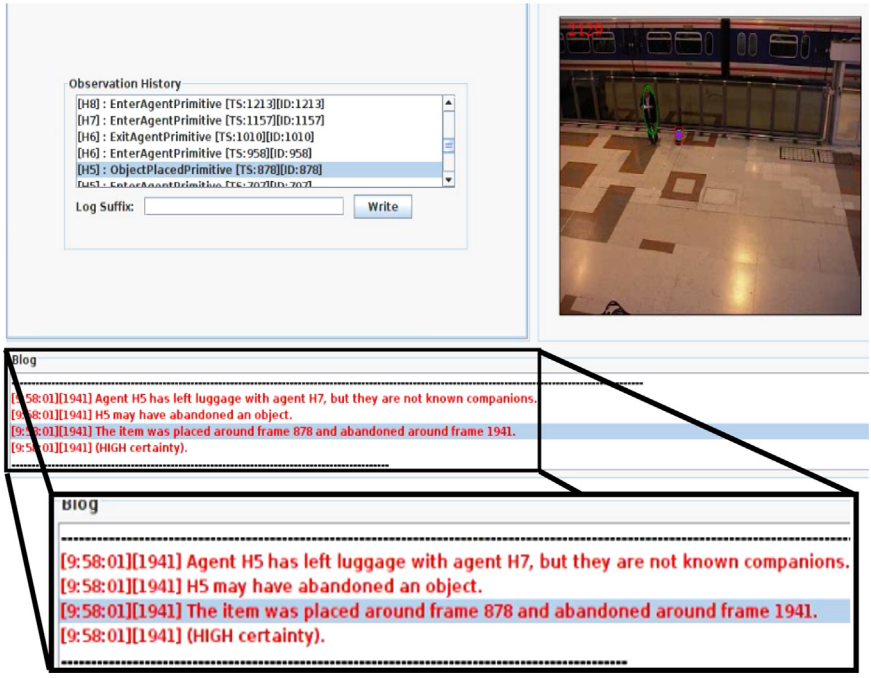

Fig. 6. Example explanation from the PETS dataset (Scenario 4). The text reads:"Agent H5 has left luggage with agent H7, but they are not known companions. H5 may have abandoned an object. The item was placed around frame 878 and abandoned around frame 1941. (High certainty)".

each root activity, an explanation template was provided containing detail 'slots'. During prediction details were extracted from the filters (e.g. agent ID, activity time) to provide a complete activity description. Example descriptions can be observed in the supplementary videos, and in the example of Fig. 6.

The results were used to generate the number of true positive (TP), false positive (FP) and false negative (FN) classifications from the PETS 2006 ground-truthed test data and our own dataset. We quote, principally, the *F*-Score, defined as the weighted average of precision and recall with range [0:1] calculated as

$$F\text{–}Score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

7.1. Person tracking

Our tracker consisted of a set of Sequential Importance Sampling (SIR) filters with re-sampling implemented on combined CPU/GPU architecture [49]. For brevity we only repeat the salient details in this paper. Output frames from this tracker have already been presented in Figs. 1 and 2. Our implementation used 100 particles to represent a person's position on the ground plane, velocity, and direction of travel. A uniform grid was generated on the ground plane using a homography transform derived from the camera calibration data using Zhang's technique [50]. For each grid point an ellipsoid was projected with height ≈ 1.8 m and diameter ≈ 0.4 m (an average for human subjects), around which a rectangular bounding box was generated within the image plane using the inverse transform. For each frame of video background subtraction was used to identify foreground pixels (blobs), and bounding boxes with $\geq 65\%$ foreground pixels were classed as person detections. A new filter/track was instantiated for each unexplained detection. Instantiating a track implied the entry of an agent into the scene, while track termination indicated departure. To address temporary occlusion (e.g. people crossing paths), particles also contained a visibility variable to indicate the person's disappearance. A Markov model was used to determine changes in

visibility state, while track termination occurred after 5 s of invisibility (full details available in [51]).

7.2. Immobile object detection

Our approach to immobile object detection was similar to other published work [49,52,53]. Background subtraction was used to identify foreground pixels using a static (empty) reference frame. Luggage objects were then identified as blobs having a real-world width/height of [0.3, 1] metres using the same technique as our person tracker. This range was chosen by manually measuring items similar to those of interest in the PETS dataset. Excluding blobs <0.3 m had the additional benefit of eliminating many small blobs created by lighting changes.

At each time-step we obtained a matrix of detected item centroids. Because immobile objects were of interest, we correlated objects between frames using a spatial threshold of 0.3 m (corresponding to the width of each grid point). We also applied an n-second persistence rule before instantiating new 'immobile object' detections, and likewise for terminations through n-seconds of absence. The temporal threshold served to reduce the number of candidate immobile objects caused by temporary lighting effects and allowed the object position to stabilize. This approach is broadly similar to [33]. In our validation $n=1$ was found to work well across both video datasets.

For both the person and object trackers, output consisted of time indexed tuples containing a unique track ID, the person/object coordinates within the image frame, and the real-world coordinates.

7.3. Extracting primitives

We detected video events from the tracker output using elementary semantic rules. These events were then correlated with primitive activities to provide input to the reasoning layer of the framework. Table 2 summarises these rules in which the parameter μ is a threshold determining how close agents must be to be considered 'grouped'. As with similar work (e.g. [54]), we draw on proxemics research from the psychology and sociology literature to determine a suitable value for μ. The term proxemics was proposed by Hall [55] and relates to the social use of space, and in particular personal space: an area with invisible boundaries surrounding an individuals body that defines a comfort zone for inter personal communications. Hall identifies a number of classifications for personal space, with inter-person distances ≤ 3.5 m broadly meeting our definition of 'grouped' individuals. We chose a mid-range value of $\mu=2$ m for our experiments.

In the evaluation video some grouped persons were occluded when entering the scene, and later split apart allowing the tracker to detect them. Because our tracker was not configured for multi-view camera tracking in our experiments (which may have alleviated this problem), we defined a 'spawning' threshold η to trigger 'split' activity primitives under these conditions. We empirically identified $\eta=2$ m as a good value from the PETS dataset, although multi-view tracking is recommended as a more robust solution to this problem in future work.

Note that object related primitives are associated with both an object and an actor. To achieve actor association we use the same approach as [5,53]: the person previously closest to the object when the primitive was detected was the activity performer. It should be highlighted that because *ObjectPlaced* and *ObjectRemoved* are associated with both person and object identifiers, one can determine that *ObjectPlaced(obj1)* and *ObjectRemoved(obj1)* refer to the same object but that *ObjectPlaced(obj1)* and *ObjectRemoved(obj2)* do not.

It should also be highlighted that it is here, at the primitive extraction layer, that a small amount of temporal information is automatically captured by virtue of the fact that a foreground blob of any kind cannot 'disappear' before it has 'appeared'. Consequently, objects and people alike can only be removed/leave after they have been placed/entered. This small amount of temporal information is gained without explicit modelling.

7.4. Parameter sensitivity

This section has described the video processing pipeline we implemented to convert raw surveillance videos into primitive activity observations. Many of the techniques employed are based on prior work with the PETS 2006 dataset, where the trend has been to hand-pick parameter values (e.g. [33,53,56]), in part because training data is not provided as part of the dataset. We have followed the same approach, but it should be noted that different parameter values could be required in alternative environments.

An advantage of our decoupled framework is that the primitive activity detectors can be easily replaced as the state-of-the-art evolves, without directly impacting inference. This allows different detectors to be integrated as required. For example, *EnterAgent* and *ExitAgent* detections could be made more robust by using spatial context to restrict their occurrence to learnt regions (as in [38]), and Lin and Sun [57] presented a more extensive group interaction algorithm for detecting forming, splitting, ignoring, following and chasing. Leach et al. [58] showed that head-pose can help identify social groups. Integrating these or other detection algorithms into the framework in future work is trivial because the primitive activity detectors are not tied to bag-of-activities inference. However, it should be noted that any such algorithms could be susceptible to the quantity of training data available.

8. Experiments

To validate our approach we performed two sets of experiments: (1) using a combination of both simulated and real video

Table 2
Primitive activity detection rules. $Pers_t$ and Obj_t are the sets of person and static luggage object detections (respectively) at time t. $pos(\kappa)_t$ is the function returning the position of person/object κ at time t. μ and η are thresholds.

Event conditions	Primitive emitted
$\exists \rho : \rho \ni Pers_{t-1} \wedge \rho \in Pers_t$	$EnterAgent(\rho)$
$\exists \rho : \rho \in Pers_{t-1} \wedge \rho \ni Pers_t$	$ExitAgent(\rho)$
if $\exists \phi : \phi \ni Obj_{t-1} \wedge \phi \in Obj_t$ then $\rho^* = \mathrm{argmin}_{\rho \in Pers_t} \lVert pos(\rho)_t - pos(\phi)_t \rVert$	$ObjectPlaced(\rho^*, \phi)$
if $\exists \phi : \phi \in Obj_{t-1} \wedge \phi \ni Obj_t$ then $\rho^* = \mathrm{argmin}_{\rho \in Pers_{t-1}} \lVert pos(\rho)_{t-1} - pos(\phi)_{t-1} \rVert$	$ObjectRemoved(\rho^*, \phi)$
$\exists \rho \neq \rho' : \{\rho, \rho'\} \subset Pers_t \cap Pers_{t-1} \wedge \lVert pos(\rho)_{t-1} - pos(\rho')_{t-1} \rVert > \mu \wedge \lVert pos(\rho)_t - pos(\rho')_t \rVert \leq \mu$	$FormGroup(\rho, \rho')$
$\exists \rho \neq \rho' : \{\rho, \rho\} \subset Pers_t \cap Pers_{t-1} \wedge \lVert pos(\rho)_t - pos(\rho')_t \rVert > \mu \wedge \lVert pos(\rho)_{t-1} - pos(\rho')_{t-1} \rVert \leq \mu$	$SplitGroup(\rho, \rho')$
$\exists \rho \neq \rho' : \{\rho, \rho'\} \subset Pers_t, \rho' \ni Pers_{t-1}, \rho \in Pers_{t-1} \wedge \mu \leq \lVert pos(\rho)_t - pos(\rho')_t \rVert < \eta$	$SplitGroup(\rho, \rho')$
$\exists \rho \neq \rho' : \{\rho, \rho'\} \subset Pers_{t-1}, \rho' \ni Pers_t, \rho \in Pers_t \wedge en \lVert pos(\rho)_{t-1} - pos(\rho')_{t-1} \rVert \leq \mu$	$SplitGroup(\rho, \rho')$

Table 3
Root activities used in the evaluation and their presence in the PETS [59] and Heriot-Watt (HW) datasets. Synthesised scenarios are indicated with an 'S' in column 3 (further details in Section 8).

Name	Description	In PETS	In HW
Passing Through 1 (PT1)	Person enters and leaves the scene	Y	Y
Passing Through 2 (PT2)	Persons enters, temporarily places luggage, then leaves the scene (with luggage)	Y	Y
Watched Item (WI)	Two people enter the scene as a group. One places luggage and leaves the scene without it. The other person remains in the scene	S	Y
Abandon Object 1 (AO1)	Two people enter the scene independently. The people temporarily form a group. One person places luggage and leaves the scene	Y	Y
Abandon Object 2 (AO2)	Person enters the scene, places luggage and leaves without it	Y	Y
Theft (Th)	Person enters the scene and places luggage. Second person enters scene and removes other agent's luggage, leaving the scene with it	N	Y
Hand-Off (HO)	Person enters the scene, places luggage and leaves. Second person enters the scene, removes luggage and leaves the scene with it	N	Y

data within a surveillance setting and (2) using simulated data to compare performance with a competing approach for video-classification.

8.1. Surveillance scenarios

Table 3 describes the seven root activities used throughout the validation. Four of these can be directly observed in the publicly available PETS 2006 video dataset [59]. This paper is the first to consider complex activities within this data and its realism and widespread use make it a good choice for future comparisons with alternative approaches. In addition to the four directly observable activities a fifth (synthetic scenario, marked with an 's' in column 3 of Table 3) was generated by merging tracking information from two separate videos. This was achieved by truncating the tracker output from S4-T5-A at video frame 1000 and joining this with the tracker output for S6-T3-H commencing with frame 1370. The use of synthetic scenarios is not uncommon in prior work (e.g. [60,61]), where it is often recognised that extending existing datasets is difficult. Finally, two further activities were defined which were not present nor synthesised in the PETS 2006 dataset, but were present within our second dataset. This second video dataset contained all activities and was gathered for the evaluation to increase the variability of activities. Termed the HW data, it included 20 instances of each activity.

We used an HMM classifier as a comparative baseline to our method, mobilising prior knowledge about the temporal order of activities to set model parameters. Different levels of HMM sensitivity were explored using three different classification strategies. The first two strategies classified observations using a ratio σ, where $\sigma=2$ implies that classifications should only be made if a model is at least twice as probable as any other. The other σ value used was 1.5. The third strategy used a threshold θ, where $\theta=0.5$ means a classification was made if the normalised probability was ≥ 0.5. σ and θ were not used concurrently.

8.1.1. Primitive activity recognition

We first discuss the recognition performance of the primitive activity detectors. The raw video data was presented to the person and object trackers, and their output was in turn presented to the primitive activity detectors. The person tracker achieved a *multiple object tracker accuracy* (MOTA) [62] of 93%.[1] Fig. 7a shows the precision of each of the six primitive activity detectors: *EnterAgent*, *ExitAgent*, *FormGroup*, *SplitGroup*, *PlaceItem* and *RemoveItem*. It is clear that the *FormGroup* detector had a low precision in the PETS data (0.33), and was slightly higher (0.62) in the HW data. This was because the PETS data only exhibits one example of the *FormGroup* activity, although two false-positives were also detected in a particularly challenging scene (S1-T1-C). Frames from this scene are shown in Fig. 8 where one can observe that four agents enter the scene in close proximity. This in itself caused problems for the person tracker, which changed the persons associated with the green and yellow ellipses, and failed to detect two agents at all.

The HW data contains eight true-positive *FormGroup* detections but five false positives are also generated during some of the *TH* behaviours. *RemoveObject* was also detected with low precision in both datasets (HW: 0.65, PETS: 0.5), while the other activities were detected with reasonable accuracy (≥ 0.74).

Fig. 7b shows that the recall of the detectors was generally high. Again, the primary exception was *FormGroup*, which had a recall of only 0.47 in the HW data. All other activities had a recall of ≥ 0.73 on both datasets. Combining the precision and recall over both datasets gave a mean recognition *F*-Score of 0.78.

These results demonstrate that our primitive activities can be detected from both video datasets, although the single-camera via makes them susceptible to target occlusions. As suggested in Section 7.3, using multiple camera views could alleviate this problem and increase detector performance.

8.1.2. Comparing model likelihood

Fig. 9a shows the probability of each root activity as the number of observations increases. After the first observation the *PT1* (passing through 1) activity is the most probable, even though all root activities can explain the observation. The reason for this is that shorter activities have a higher posterior probability $p(D|C,T,I)$. Being the shortest root activity, each *PT1* particle making a correct prediction gains a higher joint probability than other activity particles, leading ultimately to the effect observed.

At the second observation *PT1* can no longer explain the observations, while *AO2* (abandon object 2) significantly increases in probability (66% of *AO2* has been observed). At the third observation the probability of *AO2* increases further, as do the other activities also able to explain the observation. In this example *AO2* was correctly identified as the observed root activity.

To contrast these results Fig. 9b shows the baseline (HMM) probability for the same activities. The HMM models cannot distinguish between *HO* (hand-off) and *AO2*, which are identical for the first three observations when observed sequentially. After three observations the HMMs are unable to distinguish between

[1] Full validation of our tracker is available in [49].

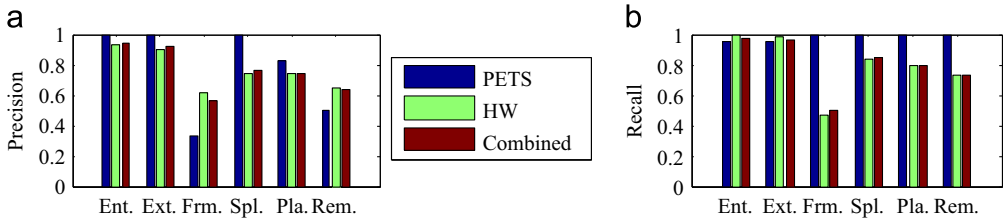

Fig. 7. (a) Primitive activity detection precision. (b) Primitive activity detection recall.

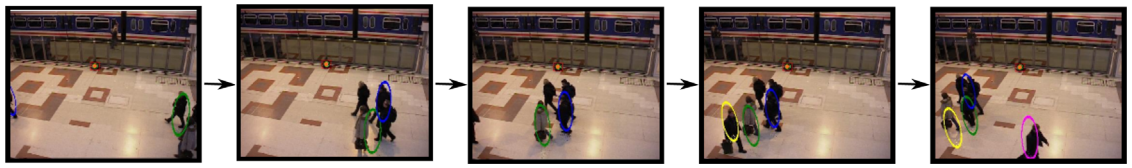

Fig. 8. Frames from PETS scene S1-T1-C. A group of agents entering the scene together and travelling as a group causes tracking errors and false *FormGroup* detections. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

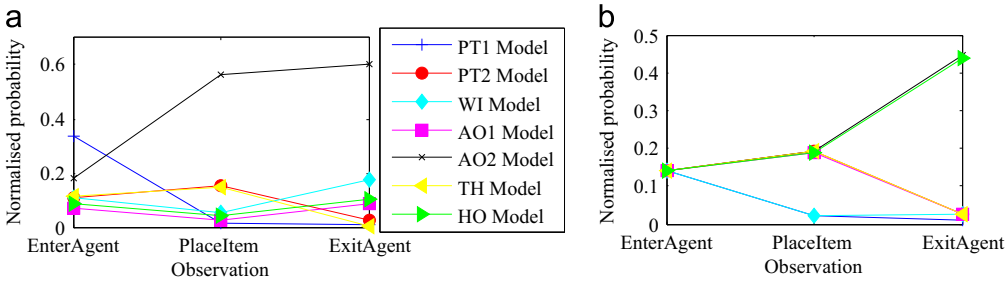

Fig. 9. Normalised root activity likelihoods while observing *AO2* (abandon object 2). (a) Bag-of-activities (b) HMMs.

these activities, preventing it from making a classification (as will be shown in the next section). Under the assumption that primitives are detected reasonably reliably, it is true that *AO2* should be more probable then *HO*, and thus our approach is fundamentally able to classify root activities that cannot be classified by the HMMs.

We next analysed model sensitivity with respect to changes to temporal order while observing the *AO1* (abandon object 1) root activity. Such changes could arise from natural variability, human adaptation, or errors encoding domain knowledge. In a distributed sensor network observations could also be delayed by unknown and/or varying degrees of latency. Fig. 10 (top) can be considered the 'control' in this experiment, where the observation sequence matched the model order. Fig. 10 (middle and bottom) show the HMM was very sensitive to a single observation arriving 'out-of-order' (the luggage item was placed at different points). Our approach was also affected by the change of order, but still converged with a high probability to the correct model, outperforming the HMM in both cases. The temporal order still had some effect on our model because some activities were common to both root activities. A person could switch their root activity at any point, and thus it is possible for a particle to ignore previous (inconsistent) observations with a low probability.

Fig. 10 sequence 1 also showed that some primitives were more informative than others. The forth observation in sequence 1 was 'FormGroup'; a primitive activity belonging only to the *AO1* root activity. This lead to a significant increase in *AO1* probability at observation 4, mimicking a high transition probability of a HMM.

8.1.3. Recognition performance

Fig. 11a shows bag-of-activities recognition *F*-Scores on our two video datasets (PETS and HW). The HW dataset contained 49,348 frames and included 20 examples of each root activity. The PETS dataset contained 75 examples in total (12,195 video frames), however, activities *TH*/*HO* (*Theft*/*Hand-off*) were absent, and most root activities were poorly represented. This is reflected in the results on the PETS data, where all root activities were detected with a very high *F*-Score. The mean *F*-Score was 0.99, with incorrect classifications being caused by the person tracker incorrectly detecting luggage items as people (a common failure mode of a real-time visual system). When the framework was applied to the more challenging HW dataset the results obtained varied. *PT*1 was still recognised with the lowest *F*-Score (0.57) and was again caused by tracking inaccuracies. The remaining root activities were all detected with an *F*-Score ≥ 0.74, and had a mean *F*-Score of 0.77.

Because of the limited nature of the PETS data, the remaining experiments focused exclusively on the more challenging HW dataset. Fig. 11b shows that our approach out-performed the HMMs for 5/7 root activities, with very poor *AO2* recognition by the HMM (mean *AO2* *F*-Score for the HMMs: 0.08). This can be explained by the HMMs inability to distinguish between two of the root activities (*AO2* and *HO*), as discussed in Section 8.1.2.

Fig. 12 shows the error range of each metric (precision, recall and *F*-Score) for the competing approaches. In part, the inability of the HMMs to recognise *AO2* can be ignored by initially focusing on the precision metric, which is the proportion of positive test results that are true positives. Our model achieved a mean precision that is significantly higher than the HMMs, at the cost of a slightly higher error range. Our approach showed significantly higher means and shorter error ranges for both recall and overall *F*-Score.[2]

[2] This remained the case even when *AO2* was excluded.

8.1.4. Comparison to the state-of-the-art

The novelty of complex activity recognition from video surveillance makes comparison difficult: no work to our knowledge generates descriptions of complex scenarios with such accuracy. Research in complex video activity recognition is not as well organised as that of primitive activity recognition. As such there are few common surveillance activities or benchmarked datasets. However, the detection of abandoned objects has received considerable attention. We therefore make a comparison against the recognition of this root activity: the "abandoned object (2)" recognition. Note that the techniques compared against are *not* recognising activities with any layered reasoning, simply recognising *metric distance from person to object*. We compare precision and recall against the benchmarked PETS data where it may be derived from the published work on this dataset.

Table 4 shows the results of this comparison with several state-of-the-art approaches (all eliminate model learning by using hand-picked parameters). *F*-Scores have been calculated for videos 1, 3, 4 and 6 in PETS, which were the same videos used in this paper. Our approach out-performs competing techniques.

8.1.5. Simulated corpus

In addition to the use of actual video datasets a simulator was developed to allow hundreds of scenarios to be generated, greatly extending test data variability. The simulator used a plan library to generate observation sequences in a similar way to prior work [63]. To enhance realism the simulator added spurious noisy observations at a rate and distribution similar to the real video event detectors. By "noise" we mean artificial low-level misclassifications of the primitive activities.

Effect of detector accuracy: We next evaluate our approach with less robust primitive detections. This might occur in more complex environments, or where parameters are estimated from insufficient training data. We used simulated observation sequences containing 0–40% noise for this analysis. Fig. 13a shows an *F*-Score of 0.97 was achieved at zero noise, which dropped by ≈ 0.1 for every 10% increase in noise. Cross-referencing these results with the video-based experiments highlights that because high-level inference has no means of recovering from missed primitive detections, it is susceptible to poor recall detectors. In simulations with 10% noise the mean *F*-Score was 0.92, while the combined video experiments (both datasets) gave an *F*-Score of 0.82. Although the simulator added noise at the same rate as the real video event detectors, recall was assumed to be 1 (no false negatives), while the real video detectors gave a mean hit-rate of only 0.8. This showed the importance of having detectors with good hit-rates, and it is likely that more robust video processing methods would be needed to maintain accuracy levels in more complex environments.

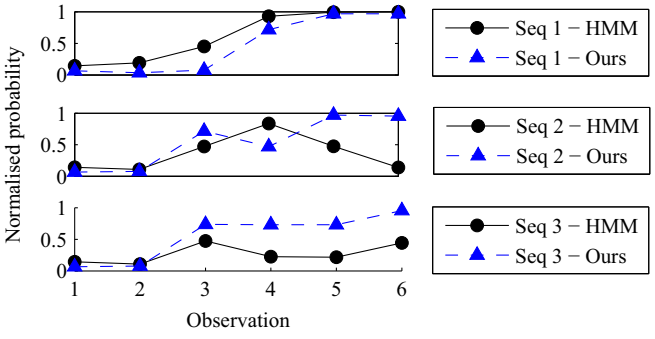

Fig. 10. Varying the *AO1* (abandon object 1) observation order affects recognition performance. (Top) Ordering matches model, (Mid/Bot.) Ordering differs from model.

Effect of novel and multi-goal activity: To analyse sensitivity to novel bags-of-activities, Fig. 13b shows recognition performance under two scenarios. When observations were constructed from the seven known root activities the mean *F*-Score was 0.92 and the mean false positive rate was 0.31%. The second scenario included instances of unknown (random) bags-of-activities in the observation streams. Encouragingly, only 13% of the unknown bags were misclassified. This gave an average *F*-Score of 0.89 with negligible effect on the FP rate.

Novel bags-of-activities could also be agents pursuing multiple root activities concurrently. We report recognition performance of concatenated and switched activity in Table 5. Concatenation is the (complete) observance of one root activity followed by another. Switching is the partial observance of two different root activities as observed when an agent aborts one to pursue another. Concatenated activity recognition gave an *F*-Score of 0.73. This performance was caused by the fact that not all particles represented the same state and thus only some of the particles were reset when a root activity completed. As a result, a subset of the particles attempted to explain all observations, reducing accuracy.

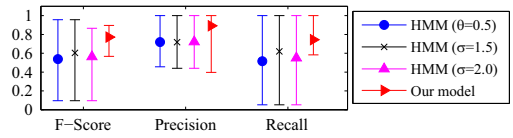

Fig. 12. Baseline comparison for recognition performance on the HW data showing the mean and range of each metric.

Table 4
Comparison of *AO2* (abandoned object 2) detection accuracy with state-of-the-art techniques.

Approach	*F*-Score
Bag-of-activities	**1.0**
Krahnstoever et al. [64]	0.86
Smith et al. [53]	0.86
Guler and Farrow [65]	0.8

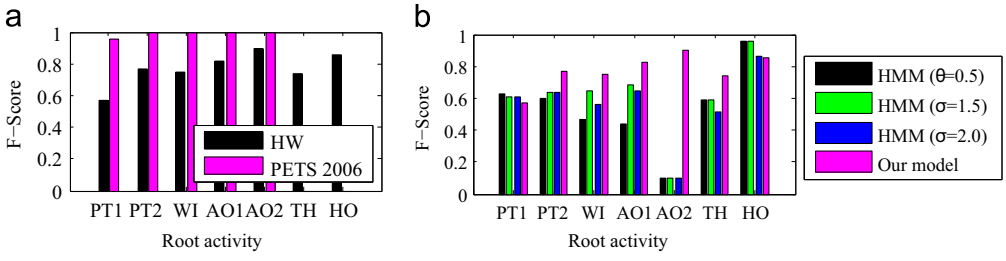

Fig. 11. (a) Recognition performance on the HW and PETS 2006 data (PETS does not contain the *Theft* (*TH*) or *Hand-off* (*HO*) activities). (b) Baseline comparison for *F*-Score on the Heriot-Watt (HW) dataset.

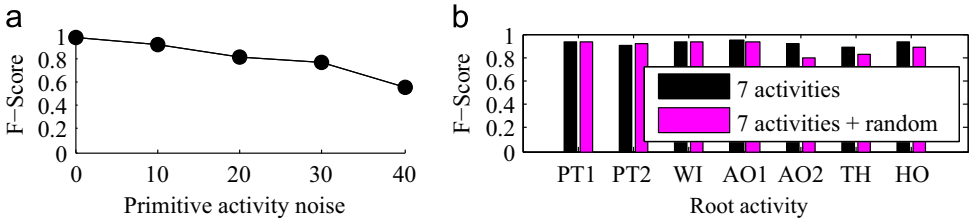

Fig. 13. (a) Effect of primitive noise on *F*-Score. (b) Activity prediction *F*-Score for the seven known root activities and an unknown (random) root activity.

Table 5
Summary of multiple goal activity performance on simulated data.

Goal-type	*F*-Score
Single-goal	0.96
Concatenated	0.73
Switched	0.68
Multi-goal	0.88

When observing switched activities the first root activity was not completed. This prevented termination detection and as a result, no predictions were made for the first activity. Predictions of the second root activity had a mean *F*-Score of 0.68. This reduced performance is a result of the limited length and large overlap of the root activities. When few primitives were observed after a root activity had switched the filters rarely converged to the new root activity before termination. Indeed, this result is consistent with work by Geib and Goldman [66] on detecting activity abandonment who remarked that the length of the evaluation activities was a significant cause of limited performance.

Run-time performance: Our final experiment in this section analysed run-time performance as the number of particles (N) was increased (Intel 2.4Ghz i5 PC with 4 GB RAM.). To isolate the number of particles the number of agents was held at six. In Fig. 14 every additional 100 particles increased total inference time by ≈ 1000 ms, and cross-referencing this with *F*-Score indicated that 200–300 particles would deliver an *F*-Score ≥ 0.8 in around 3 s when primitive activity noise was 10%. Using this analysis, we identified 220 particles per complex filter as a suitable configuration for all the experiments reported. With this configuration we found that activity alerts were made with ≈ 2.5 s frame latency, which is consistent with Fig. 14.

In our experiments video detection and tracking were performed offline, so Fig. 14 reports the run-time of the *Reasoning* and *Operator Interaction* layers of the framework (Fig. 3). However, detection and tracking can be performed in real-time and online (see for example [49]), and it should be noted that none of our bag-of-activities inference algorithms were highly parallelised. Particle filters are inherently good candidates for parallelisation, and since temporal ordering is not considered in our algorithm, it is anticipated that good performance gains could be achieved through parallelisation, which is the focus of ongoing research.

Two other factors should also be highlighted: (1) Our framework does not impose temporal thresholds on either the intervals between primitive activities, or on the overall duration of root activities. Doing so could hinder recognition of very fast or slowly evolving activities. As a consequence, activity information for all agents is maintained indefinitely. (2) The number of agents in memory changes the number of active filters and correspondingly; runtime. Some ideas for reducing memory requirements in the future are discussed in Section 9.

8.2. Video classification scenarios

A second comparison with the state-of-the-art was made against Merler et al. [21] who focus on multimedia video classification. Using the TRECVIDMED 2010 dataset[3] they address classification of 3 root activities; *assembling a shelter*, *batting a home run* and *baking a cake*. We used their reported results for the most highly correlated model vectors (primitive activities) to define the bags-of-activities for each root activity. Primitive activity detections were simulated using the average precision metrics from [21]. Miss-rates and false positive rates are not reported in [21] so we included classification errors (false positives) and missed detections into our simulated data with range $[0, 40]\%$. Each root activity was simulated 400 times per test condition. Because we simulated missed detections, we report the most probable root activity at the end of each observation sequence.

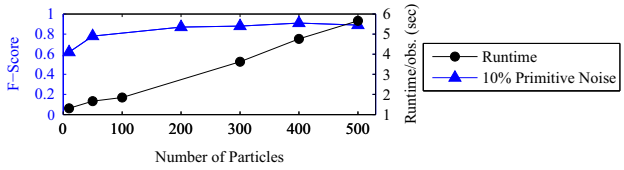

Fig. 14. Effect of number of particles on runtime and *F*-Score.

Merler et al. report average precisions of 0.36, 0.69 and 0.32 for *baking a cake*, *assembling a shelter*, and *batting a home run* respectively (Fig. 15). Because a direct comparison cannot be performed, we compared performance at a high level of classification errors / miss rate (both 40%). It is clear from Fig. 15 that our approach performs better for two activities, and offers comparable performance on the third.

It is important to highlight that [21] uses training data to learn the primitive/complex activity detectors. This has two bearings on our comparison: 1) our approach assumes that training data is unavailable so requires a higher degree of manual configuration. 2) Since the efficacy of model learning can be affected by the availability of training data it is conceivable that the performance of [21] could be improved with further training.

However, precision alone is insufficient for demonstrating performance. Fig. 15 shows that the *F*-Score gradually reduced from 1 as the classification error and miss detection rates were increased. An important observation is that the *assembling a shelter* and *batting a home run* root activities were increasingly confused with *baking a cake*. The reason for this was that the bag-of-activities for *baking a cake* was significantly smaller (by more than 50%) than the other two activities. When fewer genuine primitives were observed through missed detections, and more false primitives observed through classification errors, particles representing *baking a cake* had a higher probability of predicting an activity that was miss detected (and thus gained weight). This resulted in the same effect as that discussed in Section 8.1.2 where particles representing the smaller bag-of-activities gained higher posterior probabilities than those representing larger bags. Because root activity classification errors were drawn towards *baking a cake* the precisions of the *assembling a shelter* and *batting a home run* root activities actually started to increase again as the classification error/miss rates continued to increase. Inversely, the precision of *baking a*

[3] http://www.nist.gov/itl/iad/mig/med10.cfm.

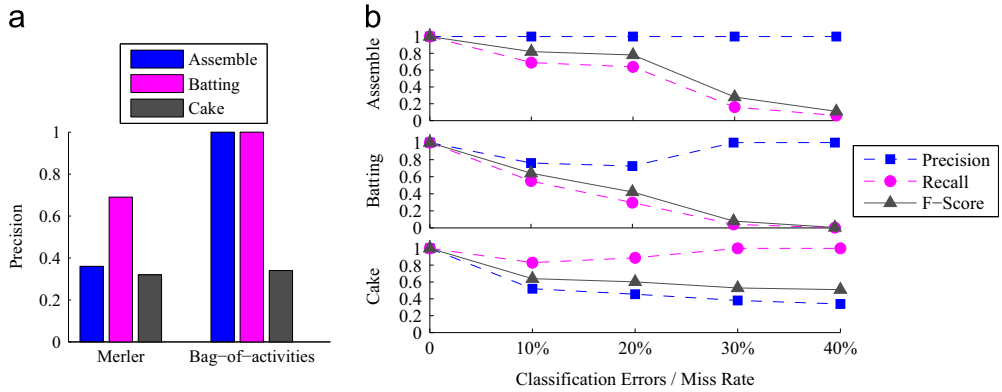

Fig. 15. (a) Comparison of root activity recognition precision (at 40% classification error/miss rate) with [21]. (b) Recognition performance as the primitive activity classification error and miss rate are increased.

cake continued to decrease, while the recall of *baking a cake* re-gained performance.

9. Discussion, conclusion and future work

This work has dealt with the issues missing in the literature on complex activity inference by developing a new probabilistic technique for data-scarce domains. We have presented both a novel generic framework and a new surveillance application that has achieved strong recognition performance on benchmarked data without requiring training/experts to define key model parameters or prepare large training datasets.

As a probabilistic inference technique, the framework has been successful in recognising complex activity. This was achieved by combining Rao–Blackwellised Particle Filters with a novel activity representation that removed the need for model learning. The validation – both synthetic and real – showed that the approach is robust to noise, giving a mean recognition *F*-Score of 0.92 (mean FP rate: 0.003) when evaluating the approach in a surveillance application using seven noisy (10%) root activities. Further validation was provided by the video-processing test-harness using publicly available and newly gathered video data. A mean *F*-Score of 0.82 (mean FP rate: 0.03) was achieved on the combined datasets which were comprised of 61,543 frames of video.

Comparing performance with HMM classifiers has highlighted several benefits to our approach: (1) HMMs cannot distinguish between activities that have the same prefix, as demonstrated by only recognising one of our root activities (*AO2*) 5% of the time. In stark contrast, our approach achieved 95% recognition. (2) By *not* modelling temporal order our approach showed less sensitivity to ordering changes while sequential models such as HMMs are particularly susceptible to 'out-of-order' observations. These could be caused by encoding errors or network latency from distributed sensors. Distributed surveillance networks are receiving growing levels of interest and thus resilience to observation latency is a highly desirable feature. (3) Our approach outperformed the HMMs with a mean increase in precision and *F*-Score of 10% and 17% respectively. In-part, this improvement was achieved by being able to determine when a root activity had been fully observed. Irrespective of activity probability, our algorithm only made classifications when it determined that all sub-components had been observed. The HMMs did not have this ability and thus were more prone to make false-positive classifications.

With respect to object-abandonment detection, the *AO*1, *AO*2 and *WI* root activities could all be individually recognised. Our approach also demonstrated success in detecting the much more challenging multi-goal activities with a mean *F*-Score of 0.71. The approach successfully recognised both multi-agent and single agent activities, while most related work only considered single agents. Real-time performance was also demonstrated, but this can only be maintained when restricting the number of agents to twenty. This limitation is caused by using combinatorial search to identify multi-agent activities.

As one would expect, good low-level detection accuracy is key to high-level inference, although the results indicate that good ($\geq 80\%$) performance can still be achieved with high primitive noise (20%).

9.1. Future work

We briefly identify some of the challenges to be addressed in future work. (1) Using combinatorial search for multi-agent activity detection causes exponential growth as the number of agents increases. One approach for limiting this complexity would be to use predictions to reduce the space of potential agent pairs. (2) An assumption of our approach is that activities do not contain repetitive components. This assumption may be overly restrictive because the real issue is with regards to components that can be repeated an infinite or unknown number of times. This is because uniform probability is assigned to elements in $C \setminus T$, which is undefined when an element can be repeated infinitely. (3) Experimental results have shown that reasonable performance can be achieved with suboptimal primitive detectors even though the approach is susceptible to missed detections. It is anticipated that modelling the recall/hit rate would improve performance in scenarios where primitive activities are frequently missed by the detectors. (4) Rather than constructing specific primitive activity detectors, it may be possible to automatically extract primitives using Hierarchical Dirichlet Processes (or other clustering techniques). The extracted primitives could then be labelled and used to define the complex activities of interest.

Conflict of interest

There are no conflicts of interest.

Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014277/1, the MOD University Defence Research Collaboration in Signal Processing, and the MOD Competition of Ideas initiative Grant number RT/COM/5/058 - A1452.

Appendix A. Supplementary material

Supplementary data associated with this paper can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2015.02.019.

References

[1] Gal Lavee, Ehud Rivlin, M. Rudzsky, Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video, IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev. 39 (5) (2009) 489–504.

[2] R.H. Baxter, D.M. Lane, N.M. Robertson, Real-time event recognition from video via a Bag-Of-Activities, in: Proceedings of the UAI Bayesian Modelling Applications Workshop, 2011.

[3] R. Baxter, D. Lane, Y. Petillot, Recognising agent behaviour during variable length activities, in: European Conference on AI, IOS Press, 2010, pp. 803–808.

[4] C. Piciarelli, G.L. Foresti, Surveillance-oriented event detection in video streams, IEEE Intell. Syst. 26 (3) (2011) 32–41.

[5] J. Ferryman, D. Hogg, J. Sochman, A. Behera, J.a. Rodriguez-Serrano, S. Worgan, L. Li, V. Leung, M. Evans, P. Cornic, S. Herbin, S. Schlenger, M. Dose, Robust abandoned object detection integrating wide area visual surveillance and social context, Pattern Recognit. Lett. 34 (7) (2013) 789–798.

[6] H. Tu, J. Allanach, S. Singh, K.R. Pattipati, P. Willett, Information integration via hierarchical and hybrid Bayesian networks, IEEE Trans. Syst., Man Cybern., Part A 36 (1) (2006) 19–33.

[7] D. Phung, T. Nguyen, S. Gupta, S. Venkatesh, Learning latent activities from social signals with hierarchical Dirichlet processes, in: Handbook on Plan, Activity, and Intent Recognition, 2014, pp. 149–174.

[8] A.A. Sodemann, M.P. Ross, B.J. Borghetti, A review of anomaly detection in automated surveillance, IEEE Trans. Syst., Man Cybern. Part C 42 (6) (2012) 1257–1272.

[9] G.E. Rawlinson, The significance of letter position in word recognition (Ph.D. thesis), Psychology Department, University of Nottingham, Nottingham, UK, 1976.

[10] H.A. Kautz, A formal theory of plan recognition and its implementation, in: Reasoning about Plans, Morgan Kaufmann, 1991, pp. 69–125.

[11] A. Sadilek, H. Kautz, Location-based reasoning about complex multi-agent behavior, J. Artif. Intell. Res. 43 (2012) 87–133.

[12] C.W. Geib, R.P. Goldman, Recognizing plans with loops represented in a lexicalized grammar, in: AAAI Conference on Artificial Intelligence, 2011, pp. 958–963.

[13] N.T. Nguyen, H.H. Bui, S. Venkatesh, G. West, Recognising and monitoring high-level behaviours in complex spatial environments, in: Computer Vision and Pattern Recognition, 2003, pp. 620–625.

[14] N.M. Oliver, A. Garg, E. Horvitz, Layered representations for learning and inferring office activity from multiple sensory channels, Comput. Vis. Image Understand. 1 (2) (2002) 163–180.

[15] K.P. Murphy, Dynamic bayesian networks: representation, inference and learning (Ph.D. thesis), 2002.

[16] C.C. Loy, T. Xiang, S. Gong, Detecting and discriminating behavioural anomalies, Pattern Recognit. 44 (1) (2011) 117–132.

[17] H.H. Bui, S. Venkatesh, Policy recognition in the abstract hidden Markov model, J. Artif. Intell. Res. 17 (2002) 451–499.

[18] N.T. Nguyen, D.Q. Phung, S. Venkatesh, H. Bui, Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models, in: Computer Vision and Pattern Recognition, vol. 2, 2005, pp. 955–960.

[19] Arnaud Doucet, Simon J. Godsill, C. Andrieu, On sequential simulation-based methods for Bayesian filtering, Stat. Comput. 10 (3) (2000) 197–208.

[20] Arnaud Doucet, N. de Freitas, K. Murphy, S. Russell, Rao–Blackwellised particle filtering for dynamic Bayesian networks, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, 2000, pp. 176–183.

[21] M. Merler, B. Huang, L. Xie, Semantic model vectors for complex video event recognition, IEEE Trans. Multimed. 14 (1) (2012) 88–101.

[22] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, A.G. Hauptmann, Complex event detection via multi-source video attributes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2627–2633.

[23] Y. Yang, M. Shah, Complex events detection using data-driven concepts, in: European Conference on Computer Vision, no. 1, 2012.

[24] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[25] G. Sukthankar, K. Sycara, Robust and efficient plan recognition for dynamic multi-agent teams (Short Paper), in: International Conference on Autonomous Agents and Multi-Agent Systems, 2008, pp. 1–4.

[26] K.R. Laviers, G. Sukthankar, Using opponent modeling to adapt team play in american football, in: Handbook on Plan, Activity, and Intent Recognition, Elsevier, 2014.

[27] D. Avrahami-Zilberbrand, G.A. Kaminka, Towards dynamic tracking of multi-agents teams: an initial report, in: Proceedings of the AAAI Workshop on Plan, Activity, and Intent Recognition, 2007.

[28] X. Qin, W. Lee, Attack plan recognition and prediction using causal networks, in: Proceedings of the 20th Annual Computer Security Applications Conference, 2004.

[29] A. Hakeem, M. Shah, Learning, detection and representation of multi-agent events in videos, Artif. Intell. 171 (8–9) (2007) 586–605.

[30] F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde, J. Ferryman, Video understanding for complex activity recognition, Mach. Vis. Appl. 18 (3) (2007) 167–188.

[31] H.H. Zhuo, Action-model based multi-agent plan recognition, in: Advances in Neural Information Processing Systems, 2012, pp. 1–9.

[32] Y. Tian, R.S. Feris, H. Liu, A. Hampapur, M.-T. Sun, Robust detection of abandoned and removed objects in complex surveillance videos, IEEE Trans. Syst., Man, Cybern., Part C: Appl. Rev. 41 (5) (2011) 565–576.

[33] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane, J. Meunier, Left-luggage detection using homographies and simple heuristics, in: Performance Evaluation in Tracking and Surveillance, 2006, pp. 51–58.

[34] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, G. Theraulaz, The walking behaviour of pedestrian social groups and its impact on crowd dynamics, PloS one 5 (4) (2010).

[35] N.M. Robertson, I.D. Reid, Automatic reasoning about causal events in surveillance video, EURASIP J. Image Video Process. (Special Is) (2011), http://dx.doi.org/10.1155/2011/530325.

[36] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. (September 2009), 1–72.

[37] H. Dee, D. Hogg, Navigational strategies in behaviour modelling, Artif. Intell. 173 (2) (2009) 329–342.

[38] F. Tung, J.S. Zelek, D.a. Clausi, Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance, Image Vis. Comput. 29 (4) (2011) 230–240.

[39] O. Arandjelović, Contextually learnt detection of unusual motion-based behaviour in crowded public spaces, in: International Symposium on Computer and Information Sciences II, 2012, pp. 403–410.

[40] F. Jiang, J. Yuan, S.a. Tsaftaris, A.K. Katsaggelos, Anomalous video event detection using spatiotemporal context, Comput. Vis. Image Understand. 115 (3) (2011) 323–333.

[41] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.

[42] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Hierarchical Dirichlet processes, J. Am. Stat. Assoc. 101 (2006) 1566–1581.

[43] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, Int. J. Comput. Vis. 79 (3) (2008) 299–318.

[44] X. Wang, X. Ma, W. Grimson, Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models, Pattern Anal. Mach. Intell. 31 (3) (2009) 539–555.

[45] Benjamin Laxton, J. Lim, D. Kriegman, Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video, in: Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[46] M.E. Bratman. Plans and Practical Reasoning. In: Intention, Plans, and Practical Reasoning, New edition, Center for the Study of Language and Information, 1999, p 28-49.

[47] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, S.A. Adelaide, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.

[48] S.S. Skiena. Combinatorial Search and Heuristic Methods. In: The Algorithm Design Manual, 2nd edition, Springer, 1998, p 230-272.

[49] W. Limprasert, A. Wallace, G. Michaelson, Real-time people tracking in a camera network, IEEE J. Emerg. Sel. Top. Circuits Syst. 3 (2) (2013) 263–271.

[50] Z. Zhang, A flexible new technique for camera calibration, IEEE Pattern Anal. Mach. Intell. 1998 (11) (2000) 1330–1334.

[51] W. Limprasert, Real-time people tracking in a camera network (Ph.D. thesis), Heriot-Watt University, 2012.

[52] F. Lv, X. Song, B. Wu, V. Kumar, S.R. Nevatia, Left luggage detection using Bayesian inference, in: PETS, 2006.

[53] K. Smith, P. Quelhas, D. Gatica-Perez, Detecting abandoned luggage items in a public space, in: Workshop on Performance Evaluation in Tracking and Surveillance (PETS'06), 2006, pp. 75–82.

[54] J.C.S. Jacques, A. Braun, J. Soldera, S.R. Musse, C.R. Jung, Understanding people motion in video sequences using Voronoi diagrams, Pattern Anal. Appl. 10 (4) (2007) 321–332.

[55] E.T. Hall, The Silent Language, Anchor, 1973.

[56] L. Li, R. Luo, W. Huang, H. Eng, Context-controlled adaptive background subtraction, in: Workshop on Performance Evaluation of Tracking and Surveillance (PETS), 2006, pp. 31–38.

[57] W. Lin, M. Sun, Group event detection with a varying number of group members for video surveillance, IEEE Trans. Circuits Syst. Video Technol. 20 (8) (2010) 1057–1067.

[58] M.J.V. Leach, R. Baxter, E.P. Sparks, N.M. Robertson, Detecting social groups in crowded surveillance videos using visual attention, in: Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 461–467.

[59] D. Thirde, L. Li, J. Ferryman, An overview of the pets 2006 dataset, in: International Workshop on Performance Evaluation of Tracking and Surveillance, 2006, pp. 47–50.

[60] X. Chai, Q. Yang, Multiple-goal recognition from low-level signals, in: Proceedings of the National Conference on Artificial Intelligence vol. 20, 2005, p. 3.

[61] D.H. Hu, X.X. Zhang, J. Yin, V.W. Zheng, Q. Yang, Abnormal activity recognition based on hdp-hmm models, in: International Joint Conference on Artificial Intelligence, 2009, pp. 1715–1720.

[62] K. Bernardin, R. Stiefelhagen, Evaluating multiple object tracking performance: the CLEAR MOT metrics, EURASIP J. Image Video Process. 1 (2008) 1–10.
[63] N. Lesh, Scalable and adaptive goal recognition (Ph.D. thesis), University of Washington, 1998.
[64] N. Krahnstoever, P. Tu, T. Sebastian, A. Perera, R. Collins, Multi-view detection and tracking of travelers and luggage in mass transit environments, in: Proceedings of the Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS), 2006.
[65] S. Guler, M.K. Farrow, Abandoned object detection in crowded places, in: Proceedings of the PETS Workshop, 2006, pp. 18–23.
[66] C.W. Geib, R.P. Goldman, Recognizing plan/goal abandonment, in: Proceedings of the International Joint Conference on Artificial Intelligence, vol. 18, 2003, pp. 1515–1517.

Rolf Baxter received his Ph.D. from Heriot-Watt University, UK, in 2012, where he is now a Research Associate in the Institute of Sensors, Signals and Systems. His research interests include Bayesian inference, machine learning and computer vision, with focus on behaviour understanding and anomaly detection.

Neil Robertson received the D.Phil. degree from Oxford University, UK, in 2006. He is the principal investigator of the Vision Lab at Heriot-Watt University and an Honorary Fellow of the School of Engineering, University of Edinburgh. His research interests include collaborative robotics, human behaviour recognition, multi-modal registration, and sensor fusion.

David Lane is the Professor of Autonomous Systems Engineering at the Ocean Systems Laboratory at Heriot-Watt University, UK. His research interests are autonomous systems, sensor processing and subsea robotics. He has published widely in the scientific literature, contributing to underwater vehicle control, servoing, docking and obstacle avoidance.